`APPARATUS AND METHOD FOR CONTROLLING A DEMULTIPLEXER AND A MULTIPLEXER USED FOR RATE MATCHING IN A MOBILE COMMUNICATION SYSTEM

PRIORITY

This application claims priority to applications entitled "Apparatus and Method for Controlling Demultiplexer and Multiplexer for Rate Matching in Mobile Communication System" filed in the Korean Industrial Property Office on July 8, 1999 and assigned Serial No. 99-27407, "Apparatus and Method for Controlling Demultiplexer and Multiplexer for Rate Matching in Mobile Communication System "filed in the Korean Industrial Property Office on July 23, 1999 and assigned Serial No. 99-30095, and "Apparatus and Method for Controlling Demultiplexer and Multiplexer for Rate Matching in Mobile Communication System "filed in the Korean Industrial Property Office on August 30, 1999 and assigned Serial No. 99-37496, the contents of all of which are hereby incorporated by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to the rate matching of a channel encoded signal, and in particular, to an apparatus and method for controlling a demultiplexer (DEMUX) and a multiplexer (MUX) used for rate matching.

2. Description of the Related Art

In general, radio communication systems, such as satellite, ISDN (Integrated

Services Digital Network), W-CDMA (Wide band-Code Division Multiple Access), UMTS (Universal Mobile Telecommunication System), and IMT (International Mobile Telecommunication)-2000 systems, channel-encode source user data with an error correction code prior to transmission, in order to increase system reliability. Typical codes used for channel encoding are convolutional codes and linear blocks code for which a single decoder is used. Lately, turbo codes, which are useful for data transmission and reception, have been suggested.

A multiple-access and multiple-channel communication system matches the number of channel encoded symbols to a given number of transmission data symbols to increase data transmission efficiency and system performance. This operation is called rate matching. Puncturing and repetition are widely performed to match the data rate of channel encoded symbols. Rate matching has recently emerged as a significant factor in UMTS for increasing data transmission efficiency in the air interface and for improving system performance.

FIG. 1 is a block diagram of an uplink transmitting device in a general mobile communication system (a UMTS system, herein).

Referring to FIG. 1, a channel encoder 110 receives frame data at predetermined TTIs (Transmission Time Intervals) which may be 10, 20, 40, or 80ms, and encodes the received frame data. And the channel encoder 110 outputs encoded symbols according to a predetermined coding rate R. The frame data size (number of information bits) is determined by a (data rate of the frame data) * (TTI). If tail bits are not considered, the number of encoded symbols are determined by the (frame data size) * (coding rate R). A 1st interleaver 120 interleaves the output of the channel encoder 110. A radio frame segmenter 130 segments interleaved symbols received from the 1st interleaver 120 into 10-ms radio frame blocks of which size is determined by (the number of encoded

symbols)/(10), wherein 10 is the radio frame length unit. A rate matcher 140 matches the data rate of a radio frame received from the radio frame segmenter 130 to a preset data rate by puncturing or repeating symbols of the radio frame. The above-described components can be provided for each service.

A MUX 150 multiplexes rate-matched radio frames from each service. A physical channel segmenter 160 segments the multiplexed radio frames received from the MUX 150 into physical channel blocks. A 2nd interleaver 170 interleaves the physical channel blocks received from the physical channel segmenter 160. A physical channel mapper 180 maps the 2nd-interleaved blocks on physical channels for transmission.

As shown in FIG. 1, the UMTS uplink transmitting device is provided with rate matchers 140. The rate matcher 140 varies in configuration depending on whether the channel encoder 110 is a convolutional encoder or a turbo encoder.

When a linear block code is used (a convolutional encoder and a single decoder are used in this case) for the channel encoder, the following requirements of rate matching should be satisfied to increase data transmission efficiency and system performance in a multiple-access/multiple-channel scheme.

1. An input symbol sequence is punctured/repeated in a predetermined periodic pattern.

2. The number of punctured symbols is minimized whereas the number of repeated symbols is maximized.

3. A uniform puncturing/repeating pattern is used to puncture/repeat encoded

symbols uniformly.

The above requirements are set on the assumption that the error sensitivity of a code symbol at any position in one frame output from a convolutional encoder is similar. Although some favorable results can be produced using the above requirements, a rate matching scheme different from the convolutional encoder should be employed when using a turbo encoder because of the different error sensitivities of symbols at different positions in one frame.

When a turbo encoder is used, it is preferred that the systematic information part of the encoded symbols is not punctured since the turbo encoder is a systematic encoder. Due to the two component encoder structure of the turbo encoder, the minimum free distance of the output code is maximized when the minimum free distance of each of the two component codes is maximized. To do so, the output symbols of the two component encoders should be punctured equally to thereby achieve optimal performance.

As described above, a distinction should be made between the information symbols and the parity symbols in the encoded symbols when a turbo encoder is used, to achieve optimal rate matching. Processing, such as channel interleaving, can be interposed between the turbo encoder and a rate matcher. Nevertheless, the distinction between information symbols and parity symbols should be preserved. However, this is impossible because all of the channel encoded symbols are randomly mixed after channel interleaving.

SUMMARY OF THE INVENTION

An object of the present invention is to provide an apparatus and method for performing rate matching separately on information symbols and parity symbols during

symbol encoding in an uplink transmitting device of a mobile communication system.

Another object of the present invention is to provide an apparatus and method for disposing a DEMUX before a rate matcher in order to separate symbol data into information symbols and parity symbols in a mobile communication system.

A further object of the present invention is to provide an apparatus and method for controlling a DEMUX and a MUX for use in rate matching in an uplink transmitting device of a mobile communication system.

Still another object of the present invention is to provide an apparatus and method for controlling a DEMUX and a MUX for use in the rate matching of a turbo-encoded signal in an uplink transmitting device of a mobile communication system.

To achieve the above and other objects, there is provided a transmitting device in a mobile communication system. In the preferred embodiments of the transmitting device, an encoder receives an information bit stream in a frame as long as an integer multiple of a predetermined size and generates an information symbol and first and second parity symbols by encoding each information bit. An interleaver sequentially arranges information symbols and the first and second parity symbols corresponding to each of the information symbols row by row in an array having a number of rows and a number of columns. The number of rows and the number of columns in the array are both integers. The interleaver reorders the columns according to a predetermined rule, reading the symbols down by column from left to right, and outputs a plurality of radio frames in a stream, each radio frame having a size determined by L/(TTI/10ms), where L is number of coded symbols. A demultiplexer demultiplexes each of the radio frames received from the interleaver to the information symbols, the first parity symbols, and the second parity symbols of the radio frame. Rate matchers bypass the information

symbols and puncture or repeat the first and second parity symbols for rate matching.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, features and advantages of the present invention will become more apparent from the following detailed description when taken in conjunction with the accompanying drawings in which:

FIG. 1 is a block diagram of an uplink transmitting device in a conventional mobile communication system;

FIG. 2 is a block diagram of an uplink transmitting device provided with a DEMUX and a MUX for rate matching, according to the preferred embodiments of the present invention;

FIG. 3 illustrates an example of turbo encoder input and turbo encoder output in the uplink transmitting device of FIG. 2;

FIG. 4 illustrates an example of 1st-interleaver input with coding rate $R = 1/3$ in the uplink transmitting device of FIG. 2;

FIGs. 5A, 5B, and 5C illustrate examples of 1st-interleaver output with $R = 1/3$ in the uplink transmitting device of FIG. 2;

FIG. 6 illustrates an example of 1st-interleaver input with $R = 1/2$ in the uplink transmitting device of FIG. 2;

FIGs. 7A, 7B, and 7C illustrate examples of 1st-interleaver output with $R = 1/2$ in the uplink transmitting device of FIG. 2;

FIGs. 8A to 8D illustrate examples of radio frame segmenter output in the uplink transmitting device of FIG. 2;

FIGs. 9A, 9B, and 9C illustrate 1st-interleaver input, 1st-interleaver output, and radio frame segmenter output according to a first embodiment of the present invention;

FIGs. 10A, 10B, and 10C illustrate 1st-interleaver input, 1st-interleaver output, and radio frame segmenter output according to a second embodiment of the present

invention;

FIGs. 11A to 11D illustrate 1^{st}-interleaver input, 1^{st}-interleaver output, and radio frame segmenter output according to a third embodiment of the present invention;

FIGs. 12A, 12B, and 12C illustrate 1^{st}-interleaver input, 1^{st}-interleaver output, and radio frame segmenter output according to a fourth embodiment of the present invention;

FIG. 13 is a block diagram of a DEMUX & MUX controlling apparatus according to an embodiment of the present invention;

FIG. 14 is a block diagram of a DEMUX & MUX controlling apparatus according to another embodiment of the present invention; and

FIG. 15 is a block diagram of a DEMUX & MUX controlling apparatus according to yet another embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will be described hereinbelow with reference to the accompanying drawings. In the following description, well-known functions or constructions are not described in detail since they would obscure the invention in unnecessary detail.

For rate matching, the UMTS uplink transmitting device of FIG. 1 has rate matcher 140 that varies in structure depending on whether channel encoder 110 is a convolutional encoder or a turbo encoder, as stated before. When a turbo encoder is used as the channel encoder 110 according to the preferred embodiments of the present invention, the rate matcher 140 is so constituted as to include a DEMUX 141, component rate matchers 142, 143, and 144, and a MUX 145, as shown in FIG. 2. The DEMUX 141 separates the output symbols of the radio frame segmenter 130 into information symbols and parity symbols and switches them to the corresponding

component rate matchers 142, 143, and 144. The MUX 145 multiplexes symbols received from the component rate matchers 142, 143, and 144 and feeds the multiplexed symbols to the MUX 150 of FIG. 1.

The uplink transmitting device shown in FIG. 2 is so constituted that the systematic information symbols of encoded symbols is not punctured in view of the fact that a turbo code is a systematic code. It is preferred that the two component encoders are connected in parallel in the turbo encoder and that the minimum free distance between final codes maximizes that of each component encoder. The consideration that the best performance can be achieved by equal puncturing of the output symbols of the two component encoders is reflected in the constitution of the uplink transmitting device in FIG. 2.

According to the preferred embodiments of the present invention, the DEMUX 141 is located between radio frame segmenter 130 and component rate matchers 142, 143, and 144, while MUX 145 is located between component rate matchers 142, 143, and 144 and MUX 150 in the uplink transmitting device.

In the embodiment of the present invention shown in Fig. 2, the DEMUX 141 and MUX 145 are synchronized with each other such that the DEMUX 141 and MUX 145 switch to the same rate matcher block (i.e., if DEMUX 141 switches to rate matcher 142 to input a symbol into the DEMUX 141, then MUX also switches to the rate matcher 142 after the input symbol has been rate matched to receive the rate matched symbol.).

The turbo code used in turbo encoder 110 of FIG. 2 is a systematic code and, thusly, can be separated into a systematic information symbol X_k and parity symbol Y_k and Z_k. For turbo encoder 110, code rate R = 1/3. Hereinafter, the systematic

information symbol will be labeled with x and the first parity symbols with y and second parity symbols with z. When R = 1/3, the relationship between the input and output of the turbo encoder 110 is shown in FIG. 3.

Referring to FIG. 3, the turbo encoder output is a sequence of an information symbol x_1, a first parity symbol y_1, a second parity symbol z_1, an information symbol x_2, a first parity symbol y_2, a second parity symbol z_2, an information symbol x_3, a first parity symbol y_3, a second parity symbol z_3, ... in this order.

The 1[st] interleaver 120 interleaves encoded symbols at a TTI (Transmission Time Interval) according to the number of input symbols. Interleaving can be considered in two steps.

First Step

1. The total number of columns is determined referring to Table 1 shown below.

2. A minimum integer R_1 is found in an equation given by

$$K_1 \leq R_1 \times C_1 \qquad \ldots\ldots (1)$$

where R_1 is the number of rows, K_1 is the length of the input block (total encoded symbols), and C_1 is the number of columns, wherein the number of columns C_1 is 1, 2, 4 or 8 according to TTIs.

3. The input symbols of the 1[st]-interleaver are sequentially arranged by rows in an rectangular array having R_1 rows and C_1 columns.

Second Step

1. Columns are reordered according to an inter-column permutation pattern $\{P_1(j)\}$ ($j = 0, 1, \ldots, C-1$) shown in Table 1. $P_1(j)$ represents the original column of a j^{th} permuted column and the pattern is derived by a bit reverse method. In the bit reverse

method, the binary bit sequence of each number is reversed, e.g., 00→00, 01→10, 10→01, and 11→11, as shown by the 40 ms TTI row in Table 1.

(Table 1)

TTI	Total number of columns	inter-column permutation patterns
10ms	1	{0}
20ms	2	{0, 1}
40ms	4	{0, 2, 1, 3}
80ms	8	{0, 4, 2, 6, 1, 5, 3, 7}

2. The 1st-interleaver output is a sequence resulting from reading the permuted $R_1 \times C_1$ array by columns. Bits that do not exist in the 1st-interleaver input are excluded from outputting by eliminating I_1 defined as

$$I_1 = R_1 \times C_1 - K_1 \qquad \text{.....}(2)$$

By interleaving using Eqs. 1 and 2, the 1st interleaver 120 outputs interleaved symbols in a similar pattern as a turbo encoder output pattern, that is, in the pattern of x, y, z, x, y, z, ... (or x, z, y, x, z, y, ... with parity symbols z and y exchanged in position).

When TTI is 10ms, the number of columns C_1 is 1. Therefore, the 1st interleaver input and the 1st interleaver output are identical.

FIG. 4 illustrates an example of 1st-interleaver input after turbo-encoding 160 input bits at R = 1/3 and the TTI=80ms. In FIG. 4, a blank rectangle denotes a system information symbol x, a rectangle marked with slant lines denotes a first parity symbol y, and a rectangle marked black denotes a second parity symbol z.

In FIG. 4, the 1st interleaver 120 sequentially receives code symbols 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ..., 160 from the turbo encoder 110. Each number represents an order of encoded symbol received from the turbo encoder 110. The numbers also indicate the order by which each of the numbers has been received in the interleaver 120 (i.e., '1' has been received first by the interleaver 120, '2' has been received second, etc.). Because of the nature of a turbo code, the 1st-interleaver input follows the pattern of x, y, z, x, y, z, x, y, z, ...

FIG. 5A illustrates an example of 1st-interleaver output when R = 1/3 and TTI = 20ms. Referring to FIG. 5A, the 1st-interleaver output sequence is 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, ..., 160 in an interleaved order in the pattern of x, z, y, x, z, y, x, z, y, ...

FIG. 5B illustrates an example of 1st-interleaver output when R = 1/3 and TTI = 40ms. Referring to FIG. 5B, the 1st-interleaver output sequence is 1, 5, 9, 13, 17, 21, 25, 29, 33, ..., 160 in an interleaved order in the pattern of x, y, z, x, y, z, x, y, z ....

FIG. 5C illustrates an example of 1st-interleaver output when R = 1/3 and TTI = 80ms. Referring to FIG. 5C, the 1st-interleaver output sequence is 1, 9, 17, 25, 33, 41, 49, 57, 65, ..., 160 in an interleaved order in the pattern of x, z, y, x, z, y, x, z, y ....

FIG. 6 illustrates an example of 1st-interleaver input after turbo encoding 160 input bits at code rate R = 1/2 and TTI= 80ms. When TTI = 10ms, the 1st-interleaver input is identical to the 1st-interleaver output. In FIG. 6, a blank rectangle denotes a system information symbol x and a rectangle marked with black dots denotes a parity symbol y.

In FIG. 6, the 1st interleaver 120 sequentially receives encoded symbols 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ..., 160 from the turbo encoder 110. Each number represents an order

of encoded symbol received from the turbo encoder 110. Because of the nature of the turbo code, the 1^{st}-interleaver input follows the pattern of x, y, x, y, x, y, ...

FIG. 7A illustrates an example of 1^{st}-interleaver output when R = 1/2 and TTI = 20ms. Referring to FIG. 7A, the 1^{st}-interleaver output sequence is 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, ..., 159, 2, 4, 6, 8, ..., 160 in an interleaved order. The first half {1, 3, 5, ..., 159} of the interleaver output is information symbols x, and the last half {2, 4, 6, ..., 160} is parity symbols y. That is, the information symbols are followed by the parity symbols in the 1^{st}-interleaver output.

FIG. 7B illustrates an example of 1^{st}-interleaver output when R = 1/2 and TTI = 40ms. Referring to FIG. 7B, the 1^{st}-interleaver output sequence is 1, 5, 9, 13, ..., 155, 159, 2, 6, 10, 14, ..., 156, 160 in an interleaved order. The first half {1, 5, 9, 13, ..., 159} of the interleaver output is information symbols x, and the last half {2, 6, 10, 14, ..., 156, 160} is parity symbols y. That is, the information symbols are followed by the parity symbols in the 1^{st}-interleaver output.

FIG. 7C illustrates an example of 1^{st}-interleaver output when R = 1/2 and TTI = 80ms. Referring to FIG. 7C, the 1^{st}-interleaver output sequence is 1, 9, 17, 25, ..., 127, 135, 143, 151, 159, 2, 10, 18, ..., 144, 152, 160 in an interleaved order. The first half {1, 9, 17, 25, ..., 143, 151, 159} of the interleaver output is information symbols x, and the last half {2, 10, 18, ..., 144, 152, 160} is parity symbols y. That is, the information symbols are followed by the parity symbols in the 1^{st}-interleaver output.

The interleaver outputs shown in FIGs. 5A, 5B, and 5C are given on the assumption that an interleaver size (= 160) is an integer multiple of TTI/10ms (= 1, 2, 4, or 8). In case an interleaver size is not an integer multiple of TTI/10ms, a different 1^{st}-interleaver output is produced.

The radio frame segmenter 130 of FIG. 2 segments a frame of 10, 20, 40, or 80ms into 10-ms radio frame blocks. Because the ratio (L/T) of an input frame size (L) to the TTI (T=TTI/10ms) of an input frame is not always an integer, the number (r) of filler bits is calculated by Eq. 3 to compensate for L/T with the filler bits (L is in units of bits or symbols). Here, $T \in \{1, 2, 4, 8\}$. If the input frame size (number of coded symbols) of the first interleaver is an integer multiple of TTI/10ms, the filler bit is not needed (r=0). If the TTI is 20ms and the input frame size is not an integer multiple of 2(TTI/10ms), the number of filler bits r is 1. If the TTI is 40ms and the input frame size is not an integer multiple of 4, the number of filler bit r can be 1, to 3. If the TTI is 80ms and the input frame size is not an integer multiple of 8, the number of filler bits can be 1 to 7. The (L+r)/T value resulting from the filler bits is defined as R(number of row).

$$r = T - (L \bmod T) \qquad \ldots\ldots (3)$$

where $r \in \{0, 1, 2, 3, \ldots T-1\}$.

$$R_i = (L_i + r_i)/T_i \qquad \ldots\ldots (4)$$

If r is not 0, the radio frame segmenter 130 inserts a filler bit into the last bit position of a corresponding frame from a $(T-r+1)^{th}$ radio frame in order to maintain a radio frame size of R. The filler bit is arbitrarily chosen as a 0 or 1. Now a description will be made of the bit-basis operation of the radio frame segmenter 130.

For description of bits prior to processing in the radio frame segmenter 130, it is assumed that the number of filler bits r has been calculated. Here, t represents the index of a radio frame, ranging from 1 through T ($1 \le t \le T$). t = 1 for the first radio frame, t = 2 for the second radio frame, and similarly, t = T for the last radio frame. Each radio frame is the same size (L+r)/T. It is assumed that the 1^{st}-interleaver output is $b_1, b_2, \ldots,$

b_L, T (=TTI/10ms) $\in$ {1, 2, 4, 8}, and the radio frame segmenter output symbols are c_1, c_2, ..., $c_{(L+r)/T}$ in a 10-ms frame. Then,

(Table 2)

output symbols of the radio frame segmenter for the first 10msec: t = 1

$c_j = b_j \qquad j = 1, 2, ..., (L+r)/T$

output symbols of the radio frame segmenter for the second 10msec: t = 2

$c_j = b_{(j+(L+r)/T)} \qquad j = 1, 2, ..., (L+r)/T$

:

output symbols of the radio frame segmenter for the $(T-r)^{th}$ 10msec: t = (T−r)

$c_j = b_{(j+(T-r-1)(L+r)/T)} \qquad j = 1, 2, ..., (L+r)/T$

output symbols of the radio frame segmenter for the $(T-r+1)^{th}$ 10msec: t = $(T-r+1)^{th}$

$c_j = b_{(j+(T-r)(L+r)/T)} \qquad j = 1, 2, ..., (L+r)/T-1$

c_j = filler_bit (0/1) $\qquad j = (L+r)/T$

:

output bits of the radio frame segmenter for the T^{th} 10msec: t = T

$c_j = b_{(j+(T-1)(L+r)/T)} \qquad j = 1, 2, ..., (L+r)/T-1$

c_j = filler_bit (0/1) $\qquad j = (L+r)/T$

The purpose of using the component rate matchers 142, 143, and 144 of FIG. 2 is to increase the data transmission efficiency and improve system performance in a multiple-access/multiple-channel system using the above-described channel encoding mechanism. Rate matching refers to control of input bit number to output bit number through puncturing when the input size is larger than the output size or repetition when the input size is smaller than the output size. The symbol puncturing or repetition is generally performed periodically but the following should be considered for rate

matching when a turbo code is used.

1. Because the turbo code is a systematic code, a systematic information symbol part of encoded symbols should be excluded from puncturing.

2. The minimum free distance between final codes preferably maximizes that of each component encoder since two component encoders are connected in parallel in a turbo encoder by definition of a turbo code. Therefore, the output symbols of the two component encoders should be equally punctured to achieve optimal performance.

In the rate matching structure shown in FIG. 2, rate matching is implemented separately for each component rate matcher. The first, second, and third component rate matchers 142, 143, and 144 subject an information symbol x, a first parity symbol y, and a second parity symbol z, respectively, to rate matching. According to a given input and output sizes, each rate matcher performs puncturing/repetition on a predetermined number of symbols. This rate matching structure is built on the assumption that the DEMUX 141 outputs x, y, z, separately. Hence, the DEMUX 141 should be able to separate a radio frame received from the radio frame segmenter 130 into symbols x, y, z in a certain order.

A description of radio frame output patterns of the radio frame segmenter 130 will be given. Radio frames are read down by columns and each column corresponds to a radio frame.

FIG. 8A illustrates an output pattern of the radio frame segmenter 130 when R = 1/3 and TTI = 10ms. Referring to FIG. 8A, a radio frame output pattern is identical to a radio frame input pattern, that is, x, y, z, x, y, z, ...

FIG. 8B illustrates an output pattern of the radio frame segmenter 130 when code rate R = 1/3 and TTI = 20ms. Referring to FIG. 8B, a first radio frame RF #1 is output in the pattern of x, z, y, x, z, y, ... and a second radio frame RF #2 is output in a radio frame pattern of …, x, y, x, z, y, x, z, ... The output patterns correspond to the output from the 1 st interleaver shown in Fig. 5A.

FIG. 8C illustrates an output pattern of the radio frame segmenter 130 when R = 1/3 and TTI = 40ms. Referring to FIG. 8C, a first radio frame RF #1 is output in the pattern of ..., x, y, z, x, y, z, ..., a second radio frame RF #2 in the pattern of …, z, x, y, z, x, y, ..., a third radio frame RF #3 in the pattern of …, y, z, x, y, z, x, ..., and a fourth radio frame RF #4 in the pattern of …, x, y, z, x, y, z, ... The output patterns correspond to the output from the 1 st interleaver shown in Fig. 5B.

FIG. 8D illustrates an output pattern of the radio frame segmenter 130 when R = 1/3 and TTI = 80ms. Referring to FIG. 8D, a first radio frame RF #1 is output in the pattern of …, x, z, y, x, z, y, ..., a second radio frame RF #2 in the pattern of …, y, x, z, y, x, z, ..., a third radio frame RF #3 in the pattern of …, z, y, x, z, y, x, ..., a fourth radio frame RF #4 in the pattern of …, x, z, y, x, z, y, ..., a fifth radio frame RF #5 in the pattern of …, y, x, z, y, x, z, ..., a sixth radio frame #6 in the pattern of …, z, y, x, z, y, x, a seventh radio frame RF #7 in the pattern of …, x, z, y, x, z, y, ..., and an eighth radio frame RF #8 in the pattern of …, y, x, z, y, x, z, ... The output patterns correspond to the output from the 1 st interleaver shown in Fig. 5C.

Output patterns of the radio frame segmenter 130 have a certain regularity. Each radio frame pattern with the same TTI has a different initial symbol x, y, or z but has the same symbol repeating pattern. For TTIs = 10ms and 40ms, symbols are repeated in the pattern of …, x, y, z, x, y, z, ..., and for TTIs = 20ms and 80ms, symbols are repeated in the pattern of x, z, y, x, z, y, ...

The radio frames in the above cases are free of a filler bit. This is because the input size is an integer multiple of TTI/10ms. When filler bits are to be inserted, radio frames have different patterns from the above-described patterns. The first through fourth embodiments as described below pertain to insertion of filler bits.

First Embodiment

FIGs. 9A, 9B, and 9C illustrate 1st-interleaver input, 1st-interleaver output, and radio frame segmenter output according to a first embodiment of the present invention.

If the input of the 1st interleaver 120 for TTI = 80ms is given in FIG. 9A, it is interleaved by columns according to an interleaving rule of the 1st interleaver 120, as shown in FIG. 9B. Then, symbols are read down each column starting from the left to the right column in the array of FIG. 9B. The resulting 1st-interleaver output (i.e., the radio segmenter input) is x, z, y, x, z, y, x, z, y, z, y, x, z, y, x, z, y, x, y, x, z, y, x, z, y, x, z, x, z, y, x, z, y, x, z, y. The output of the radio frame segmenter 130 results from adding filler bits to the radio frame segmenter input.

In the first embodiment, the filler bits are 0s. In the first embodiment of the present invention, the radio frame segmenter 130 outputs the symbols received from the interleaver 120 in a such way that the all of the filler bits are placed towards the end of the last row, as shown in Fig. 9C. In Fig. 9B, the last positions in the second, fourth, sixth and eight columns are empty. Instead of filling those positions with filler bits, the next symbol coming after the empty position is used to fill the empty position. For example, to fill the last position in the second column, the 'z' symbol from the first position in the third column is moved in to the empty position in the second column. The position previously occupied by the 'z' symbol is now occupied by the 'y' symbol which came after the 'z' symbol in the third column. Basically the positions of the

symbols have been pushed up by one position. This process is repeated to fill the empty position in the fourth column, and so on. However, the last positions in the last four columns (i.e., column #5, 6, 7 and 8) are filled with the filler bits so that the filler bits are pushed towards the end of the last row, as shown in Fig. 9C. Symbols in the array of FIG. 9C are read column by column and each column represents one radio frame. As shown in FIG. 9C, each radio frame has a different initial symbol but follows the same symbol repeating pattern of x, z, y, except for radio frames 4 and 6 because of the position shifting. However, the repeating patterns for the radio frames 4 and 6, which are shown below in Table 15, can be used. The patterns in the radio frames follow the predetermined repeating patterns shown in Table 15 except for the tail ends of certain radio frames. In those cases, the tail ends are ignored and treated as if the tail ends follow the predetermined repeating patterns shown in Table 15 and are rated matched according to the predetermined repeating patterns. That is, the radio frames have different initial symbols in the filler bit inserting case, as compared to the filler bit-free case.

Although filler bits are inserted, radio frames may have the same initial symbols as those in the filler bit-free case. An example of such a case using three filler bits for TTI = 40ms will be described.

FIGs. 10A and 10B illustrate 1st-interleaver input, 1st-interleaver output, and radio frame segmenter output according to the first embodiment.

If the input of the 1st interleaver 120 for TTI = 40ms is given in FIG. 10A, it is interleaved by columns according to an interleaving rule of the 1st interleaver 120 as shown in FIG. 10B. The resulting 1st-interleaver output (i.e., the radio segmenter input) is x, y, z, x, y, z, z, x, y, z, x, y, z, x, y, z, x, y, z, x, y. The output of the radio frame segmenter 130 shown in FIG. 10C results from adding filler bits to the radio frame

segmenter input.

The filler bits are 0s. Symbols in the array of FIG. 10C are read column by column and each column represents one radio frame. As shown in FIG. 10C, each radio frame has a different initial symbol but follows the same symbol repetition pattern of ..., x, y, z, ... That is, the radio frames have the same initial symbols in this filler bit inserting case as those in the filler bit-free case.

The initial symbol of each radio frame is determined by a TTI and the number of filler bits added by the radio frame segmenter 130. Herein below, initial symbols in all possible cases will be described. Tables 3 to 6 list initial symbols for TTIs = 10, 20, 40, and 80ms, respectively, when the radio frame segmenter 130 outputs radio frames RF#1, RF #2, RF #3, RF #4, RF #5, RF #6, RF #7, and RF #8 sequentially.

(Table 3)

TTL = 10ms

total number of filler bits	Initial symbol of
	RF #1
0	x

(Table 4)

TTL = 20ms

total number of filler bits	Initial symbol of	
	RF #1	RF #2
0, 1	X	y

In Table 4, since the 1st interleaver 120 leaves the columns intact, positions are

not changed when one filler bit is used. Consequently, the initial symbols are the same as those in the filler bit-free case.

(Table 5)

TTL = 40ms

total number of filler bits	Initial symbol of			
	RF #1	RF #2	RF #3	RF #4
0, 1, 3	x	z	y	x
2	x	z	z	x

When one or three filler bits are used, the number of symbols in each column before interleaving is equal to that of symbols in the column of the same index after interleaving. Therefore, the initial symbols are the same as those in the filler bit-free case. If two filler bits are used, the number of symbols in each column before interleaving is different from that of symbols in the column of the same index after interleaving. Therefore, the initial symbols are different from those in the filler bit-free case.

(Table 6)

TTL = 80ms

total number of filler bits	initial symbol of							
	RF #1	RF #2	RF #3	RF #4	RF #5	RF #6	RF #7	RF #8
0, 1, 7	x	y	z	x	Y	z	x	y
2, 3	x	y	z	x	X	y	z	y
4	x	y	y	z	Z	y	z	y
5, 6	x	y	y	z	X	z	x	y

When one or seven filler bits are used, the number of symbols in each column before interleaving is equal to that of symbols in the column of the same index after interleaving. Therefore, the initial symbols are the same as those in the filler bit-free case. If two, three, four, five, or six filler bits are used, the number of symbols in each column before interleaving is different from that of symbols in the column of the same index after interleaving. Therefore, the initial symbols are different from those in the filler bit-free case.

As noted from the above tables, symbols are repeated in the pattern of x, y, z, x, y, z, for TTIs = 10ms and 40ms, whereas symbols are repeated in the pattern of x, z, y, x, z, y, for TTIs = 20ms and 80ms.

Therefore, given a TTI and the number of filler bits to be inserted by the radio frame segmenter 130, the DEMUX 141 demultiplexes 1st-interleaver output in the above-described manner.

Second Embodiment

FIGs. 11A to 11D illustrate 1st-interleaver input, 1st-interleaver output, and radio frame segmenter output according to a second embodiment of the present invention. The second embodiment is different from the first embodiments in that filler bits are inserted by the 1st interleaver 120 instead of the radio frame segmenter 130. Instead of pushing the filler bit positions to the end of the last row, as in the first embodiment (i.e., FIG. 9C), the interleaver 120 fills the empty positions with filler bits, as shown in FIG. 11C. In terms of initial symbols and repeating patterns, this case is the same as the typical filler bit-free case.

If the input of the 1st interleaver 120 for TTI = 80ms is given as in FIG. 11A, it is interleaved by columns according to an interleaving rule of the 1st interleaver 120 as

shown in FIG. 11B. Then, filler bits are inserted to the array of FIG. 11B as shown in FIG. 11C. Here, the filler bits are 0s. Therefore, the 1st-interleaver output, i.e., the radio frame segmenter input is a sequence of x, z, y, x, z, y, z, y, 0, z, y, x, z, y, x, z, y, x, 0, y, x, z, y, x, z, y, x, z, 0, x, z, y, x, z, y, x, z, y, 0. The output of the radio frame segmenter 130 is shown in FIG. 11D.

The symbols in the array of FIG. 11D are read down by column from left to right and each column is a radio frame. As shown in FIG. 11D, each radio frame follows the same repeating pattern of x, z, y with a different initial symbol. As noted from FIGs. 11A to 11D, the initial symbols are the same as those in the general filler bit-free case.

The initial symbol of each radio frame is determined by a TTI. Tables 7 to 10 list initial symbols for TTIs = 10, 20, 40, and 80ms, respectively, when the radio frame segmenter 130 outputs radio frames RF#1, RF #2, RF #3, RF #4, RF #5, RF #6, RF #7, and RF #8 sequentially. The initial symbols of the radio frames in the second embodiment are independent of the total number of the filler bits, as shown below; however, in the first embodiment, the initial symbols of the radio frames are dependent on the total number of the filler bits.

(Table 7)

TTI = 10ms

initial symbol of
RF #1
x

(Table 8)

TTI = 20ms

initial symbol of	
RF #1	RF #2
x	y

(Table 9)

TTI = 40ms

initial symbol			
RF #1	RF #2	RF #3	RF #4
x	z	y	x

(Table 10)

TTI = 80ms

initial symbol of							
RF #1	RF #2	RF #3	RF #4	RF #5	RF #6	RF #7	RF #8
X	y	z	x	y	z	x	y

As noted from the above tables, symbols are repeated in the pattern of x, y, z, x, y, z, for TTIs = 10ms and 40ms, whereas symbols are repeated in the pattern of x, z, y, x, z, y, for TTIs = 20ms and 80ms.

Therefore, given a TTI, the DEMUX 141 demultiplexes 1st-interleaver output in the above-described manner.

<u>Third Embodiment</u>

FIGs. 12A, 12B, and 12C illustrate 1st-interleaver input, 1st-interleaver output, and radio frame segmenter output according to a third embodiment of the present invention. The third embodiment is different from the second embodiments in that a

controller (host) designates filler bit insertion positions and the radio frame segmenter 130 inserts the filler bits in the designated positions. In terms of initial symbols and repeating patterns, this case is the same as the typical filler bit-free case.

If the input of the 1st interleaver 120 for TTI = 80ms is given in FIG. 12A, it is interleaved by columns according to an interleaving rule of the 1st interleaver 120 as shown in FIG. 12B. Therefore, the 1st-interleaver output, i.e., the radio frame segmenter input is a sequence of x, z, y, x, z, y, x, z, y, z, y, x, z, y, x, z, y, x, y, x, z, y, x, z, y, x, z, x, z, y, x, z, y, x, z, y. A controller (host) designates filler bit insertion positions and then the radio frame segmenter 130 inserts the filler bits in the designated positions as shown in FIG. 12C.

In this embodiment, the filler bits are 0s. The symbols in the array of FIG. 12C are read down column by column from left to right and each column is a radio frame. As shown in FIG. 12C, each radio frame follows the same repeating pattern of x, z, y with a different initial symbol. As noted from FIGs. 12A, 12B, and 12C, initial symbols are the same as those in the general filler bit-free case.

The initial symbol of each radio frame is determined by a TTI. Tables 11 to 14 list initial symbols for TTIs = 10, 20, 40, and 80ms, respectively, when the radio frame segmenter 130 outputs radio frames RF#1, RF #2, RF #3, RF #4, RF #5, RF #6, RF #7, and RF #8 sequentially. The initial symbols of the radio frames in the third embodiment are independent of the total number of the filler bits, as shown below.

(Table 11)

TTI = 10ms

initial symbol of

RF #1
x

(Table 12)

TTI = 20ms

initial symbol of	
RF #1	RF #2
X	y

(Table 13)

TTL = 40ms

initial symbol			
RF #1	RF #2	RF #3	RF #4
X	z	y	x

(Table 14)

TTL = 80ms

initial symbol of							
RF #1	RF #2	RF #3	RF #4	RF #5	RF #6	RF #7	RF #8
x	y	z	x	y	z	x	y

As noted from the above tables, symbols are repeated in the pattern of x, y, z, x, y, z, for TTIs = 10ms and 40ms, whereas symbols are repeated in the pattern of x, z, y, x, z, y, for TTIs = 20ms and 80ms.

Given a TTI, the DEMUX 141 demultiplexes 1st-interleaver output in the above-described manner.

Returning to FIG. 2, the DEMUX 141 demultiplexes a radio frame received from the radio frame segmenter 130 into its symbols x, y, z, according to a switching rule. The switching rule is determined by a TTI and the number of filler bits used by the radio frame segmenter 130 in the first embodiment and a TTI in the second and third embodiments. The symbols are repeated in a certain pattern. The repeating patterns for the embodiments are tabulated in Tables 15 and 16. In the tables, N/A indicates "not applicable".

(Table 15)

For First Embodiment

TTI	total number of filler bits	Switching rules (repeating patterns)							
		RF #1	RF #2	RF #3	RF #4	RF #5	RF #6	RF #7	RF #8
10ms	0	x, y, z	N/A	N/A	N/A	N/A	N/A	N/A	N/A
20ms	0, 1	x, z, y	y, x, z	N/A	N/A	N/A	N/A	N/A	N/A
40ms	0, 1, 3	x, y, z	z, x, y	y, z, x	x, y, z	N/A	N/A	N/A	N/A
	2	x, y, z	z, x, y	z, x, y	x, y, z	N/A	N/A	N/A	N/A
80ms	0, 1, 7	x, z, y	y, x, z	z, y, x	x, z, y	y, x, z	z, y, x	x, z, y	y, x, z
	2, 3	x, z, y	y, x, z	z, y, x	x, z, y	x, z, y	y, x, z	z, y, x	y, x, z
	4	x, z, y	y, x, z	y, x, z	z, y, x	z, y, x	y, x, z	z, y, x	y, x, z
	5, 6	x, z, y	y, x, z	y, x, z	z, y, x	x, z, y	z, y, x	x, z, y	y, x, z

(Table 16)

For Second and Third Embodiments

TTI	Switching rules (repeating patterns)							
	RF #1	RF #2	RF #3	RF #4	RF #5	RF #6	RF #7	RF #8
10ms	x, y, z	N/A	N/A	N/A	N/A	N/A	N/A	N/A
20ms	x, z, y	y, x, z	N/A	N/A	N/A	N/A	N/A	N/A
40ms	x, y, z	z, x, y	y, z, x	x, y, z	N/A	N/A	N/A	N/A
80ms	x, z, y	y, x, z	z, y, x	x, z, y	y, x, z	z, y, x	x, z, y	y, x, z

If two filler bits are used for TTI = 40ms in the first and second embodiments, the switching patterns in the DEMUX 141 are x, y, z, x, y, z for the first radio frame, z, x, y, z, x, y for the second radio frame, z, x, y, z, x, y for the third radio frame, and x, y, z, x, y, z for the fourth radio frame.

In the second and third embodiments, the initial symbol of each radio frame only needs to be given because the repeating patterns are already predetermined based on the TTI. However, in the first embodiment, the total number of the filler bits also needs to be given in addition to the other information. Tables 17–19 reflect that difference between the embodiments.

(Table 17)

For First Embodiment

TTI	total number of filler bits	Initial symbol of							
		RF #1	RF #2	RF #3	RF #4	RF #5	RF #6	RF #7	RF #8
10ms	0	x	N/A	N/A	N/A	N/A	N/A	N/A	N/A
20ms	0, 1	x	y	N/A	N/A	N/A	N/A	N/A	N/A
40ms	0, 1, 3	x	z	y	x	N/A	N/A	N/A	N/A
	2	x	z	z	x	N/A	N/A	N/A	N/A
80ms	0, 1, 7	x	y	z	x	y	z	x	y
	2, 3	x	y	z	x	x	y	z	y
	4	x	y	y	z	z	y	z	y
	5, 6	x	y	y	z	x	z	x	y

(Table 18)

For Second and Third Embodiments

TTI	initial symbol of							
	RF #1	RF #2	RF #3	RF #4	RF #5	RF #6	RF #7	RF #8
10ms	x, y, z	N/A	N/A	N/A	N/A	N/A	N/A	N/A
20ms	x, z, y	y, x, z	N/A	N/A	N/A	N/A	N/A	N/A

40ms	x, y, z	z, x, y	y, z, x	x, y, z	N/A	N/A	N/A	N/A
80ms	x, z, y	y, x, z	z, y, x	x, z, y	y, x, z	z, y, x	x, z, y	y, x, z

(Table 19)

Repeating Patterns

TTI	Repeating patterns
10ms, 40ms	..., x, y, z, x, y, z, ...
20ms, 80ms	..., x, z, y, x, z, y, ...

Referring to FIG. 2 again, the MUX 145 multiplexes three streams received from the component rate matchers 142, 143, and 144 to one stream, to thereby generate a rate-matched radio frame with the same symbol pattern as before rate matching. Because this MUX 145 is the counterpart of the DEMUX 141, it switches according to the same switching patterns.

FIG. 13 is a block diagram of a DEMUX and MUX controlling apparatus according to the first embodiment of the present invention.

Referring to FIG. 13, upon receipt of a TTI, the total number of the filler bits, and a radio frame length from the host 200, the controller 210 feeds the TTI, the total number of the filler bits, and the radio frame index of a current radio frame to the memory 220 (see Table 17) and receives the initial symbol of the current radio frame from the memory 220. The controller 210 controls the switching operations of the DEMUX 141 and the MUX 145 based on the initial symbol and a repeating/puncturing pattern determined by the TTI. The DEMUX 141 separates the current radio frame symbols into input for the corresponding component rate matchers and the MUX 145 multiplexes the output symbols of the rate matchers to

a radio frame. Here, the DEMUX 141 separates an information symbol, a first parity symbol, and a second parity symbol from a radio frame stream received from the radio frame segmenter 130. The component rate matchers 142, 143, and 144 rate match the information symbol, the first parity symbol, and the second parity symbol from the DEMUX 141, respectively, by puncturing or repetition. The component rate matcher 142 just bypasses the received information symbols without real puncturing, whereas the component rate matchers 143 and 144 puncture the received parity symbols according to a preset pattern which is determined by the ratio of the number of input symbols to the number of output symbols. In most of the real cases, the component rate matchers 143 and 144 just bypass the received parity symbols without real repetition except heavy repetition of the encoded symbols, whereas the component rate matcher 142 repeats the received information symbols according to a preset pattern determined by the ratio of the number of input symbols to the number of output symbols.

The MUX 145 multiplexes the symbols received from the component rate matchers 142, 143, and 144 to one stream according to the same switching pattern as used in the DEMUX 141.

FIG. 14 is a block diagram of a DEMUX and MUX controlling apparatus according to the second embodiment of the present invention.

Referring to FIG. 14, upon receipt of a TTI and a radio frame length from the host 200, the controller 210 feeds the TTI, the total number of filler bits, and the radio frame index of a current radio frame to memory 220 (see Table 17) and receives the initial symbol of the current radio frame from memory 220. The number of filler bits is determined by the controller 210 based on the TTI and the frame length in the same manner as used in the radio frame segmenter. Then, the controller 210 controls the

switching operations of the DEMUX 141 and the MUX 145 based on the initial symbol and a repeating/puncturing pattern determined by the TTI. The DEMUX 141 separates the current radio frame symbols into component rate matchers input and the MUX 145 multiplexes the output symbols of the rate-matchers to a radio frame. Here, the DEMUX 141 separates an information symbol, a first parity symbol, and a second parity symbol from a radio frame stream received from the radio frame segmenter 130.

The component rate matchers 142, 143, and 144 rate match the information symbol, the first parity symbol, and the second parity symbol from the DEMUX 141, respectively, by puncturing or repetition. The component rate matcher 142 just bypasses the received information symbol without real puncturing, whereas component rate matchers 143 and 144 puncture the received parity symbols according to a preset pattern determined by the ratio of the number of input symbols to the number of output symbols. In most of the real cases, the component rate matchers 143 and 144 just bypass the received parity symbols without real repetition except heavy repetition of the encoded symbols, whereas the component rate matcher 142 repeats the received information symbols according to a preset pattern determined by the ratio of the number of input symbols to the number of output symbols. The MUX 145 multiplexes the symbols received from the component rate matchers 142, 143, and 144 to one stream according to the same switching pattern as used in the DEMUX 141.

FIG. 15 is a block diagram of a DEMUX and MUX controlling apparatus according to the third embodiment of the present invention.

Referring to FIG. 15, upon receipt of a TTI and a radio frame length from the host 200, the controller 210 feeds the TTI and the radio frame index of a current radio frame to memory 220 (see Table 18) and receives the initial symbol of the current radio frame from memory 220. Then, the controller 210 controls the switching operations of

the DEMUX 141 and the MUX 145 based on the initial symbol and a repeating/puncturing pattern determined by the TTI. The DEMUX 141 separates the current radio frame symbols into input for the component rate matchers and the MUX 145 multiplexes the output symbols of the rate matchers to a radio frame. Here, the DEMUX 141 separates an information symbol, a first parity symbol, and a second parity symbol from a radio frame stream received from the radio frame segmenter 130. The component rate matchers 142, 143, and 144 rate match the information symbol, the first parity symbol, and the second parity symbol from the DEMUX 141, respectively, by puncturing or repetition. The component rate matcher 142 just bypasses the received information symbol without real rate puncturing, whereas component rate matchers 143 and 144 puncture or repeat the received parity symbols according to a pattern preset determined by the ratio of the number of input symbols to the number of output symbols. The MUX 145 multiplexes the symbols received from the component rate matchers 142, 143, and 145 to one stream according to the same switching pattern as used in the DEMUX 141. In most of the real cases, the component rate matchers 143 and 144 just bypass the received parity symbols without real repetition except heavy repetition of the encoded symbols, whereas the component rate matcher 142 repeats the received information symbols according to a preset pattern determined by the ratio of the number of input symbols to the number of output symbols.

As described above, the present invention is advantageous in that effective rate matching can be performed by adding a DEMUX before a rate matching unit to separate an information symbol and parity symbols of the encoded symbols when the information symbol is not to be punctured for rate matching in an uplink transmitter in a mobile communication system.

While the invention has been shown and described with reference to certain preferred embodiments thereof, it will be understood by those skilled in the art that

various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims.